REMARKS

Claims 11-34 are pending, with claims 11, 16, 21, 25 and 33 being independent. Reconsideration and allowance of the above-referenced application are respectfully requested.

Allowable Subject Matter

The indication of allowable subject matter in claim 31 is acknowledged and appreciated. The claim is retained, and the right to rewrite this claim in independent form at a later date is reserved.

Claim Rejections

Claims 11-13, 15-18, 20, 21, 23-27, 29, 30 and 34 stand rejected under 35 U.S.C. 102(b) as allegedly being anticipated by Lock Reservation (herein Koseki). Claim 33 stands rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Koseki in view Transactional Memory (herein Moss). Claims 14, 19, 22, 28 and 32 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Koseki in view of alleged "Common Art". These contentions are respectfully traversed.

Examiner Johnson is thanked for the interview, which was conducted with Mr. Hunter, on October 23, 2007. During the interview, claims 11, 13, 16, 18, 21, 23, 25, 33 and 34, and the Koseki reference were discussed. Mr. Hunter argued that Koseki does not describe the use of processor speculation, machine instructions that are speculatively executed, a machine

instruction that controls speculative execution in a processor, or mis-speculation, as claimed. Agreement was not reached.

Independent claim 11 recites, "generating parallel processes in a data processing machine; effecting synchronization between the parallel processes using processor speculation in the data processing machine to speculatively execute one or more instructions that read-modify-write a lock variable associated with a critical section and end speculation before performing the critical section; and providing output resulting from the synchronized parallel processes." (Emphasis added.) The Office cites to the Abstract of Koseki as allegedly teaching this subject matter, but Koseki's Abstract merely states:

> This paper presents a novel algorithm called lock reservation. It exploits thread locality of Java locks, which claims that the locking sequence of a Java lock contains a very long repetition of a specific thread. The algorithm allows locks to be reserved for threads. When a thread attempts to acquire a lock, it can do without any atomic operation if the lock is reserved for the thread. Otherwise, it cancels the reservation and falls back to a conventional locking algorithm.

See Koseki at Abstract. Nothing in this portion of Koseki, or any other portion of Koseki, teaches or suggests using processor speculation to speculatively read-modify-write a lock variable. The Office appears to take the position that anytime a processor performs operations that may later be discarded, this is in some sense "speculative" processing, and thus can be considered using processor speculation, as claimed.

This position should be reconsidered as it contradicts the plain meaning of the claim language, in light of the specification. For example, as described in the specification:

> The data processing machine may include an out-of-order processor/processing system that provides speculative execution of machine instructions, and this processor speculation capability is exposed to program control. Using processor speculation to implement synchronization among parallel processes may provide a significant advantage in that, if a critical section of a program happens to be uncontended at runtime (e.g., only one of the processes happens to need the critical section at a given time), then the overhead of traditional locking may be eliminated.

See Specification at page 4, paragraph 13. While it is not appropriate to read limitations into the claims from the specification, it is also not appropriate to disregard the specification entirely when assessing the scope of the claims. It is well established law that the words used in the claims must be considered in context, and the specification is the primary source of that context.

Claim 11 expressly recites, "using processor speculation in the data processing machine to speculatively execute one or more instructions that read-modify-write a lock variable" (emphasis added). This clearly refers to the use of the speculative execution capabilities of modern processors. Koseki does not in any way teach or suggest such speculative execution of one or more instructions that read-modify-write a lock variable.

Moreover, assuming for the sake of argument that Koseki's lock reservation scheme constitutes speculative execution, which is not conceded, Koseki still fails to teach the claimed subject matter, which requires ending speculation before performing the critical section. Koseki clearly describes that a thread's reservation can be cancelled even when a thread is executing the critical section. *See* Koseki at page 133. In Koseki, the "speculation" (as interpreted by the Office) continues even after the critical section has started being performed. *See* Koseki at page

133-134. Thus, even under the Office's current interpretation of the claimed speculative execution, Koseki still fails to meet all the elements of claim 11.

For all of the above reasons, the rejection of claim 11 suffers from clear legal or factual deficiencies. Thus, claim 11 should be in condition for allowance.

Independent claim 16 recites, "generating parallel processes in a data processing machine; effecting synchronization between the parallel processes using processor speculation in the data processing machine to speculatively execute one or more instructions that read-modify-write a lock variable associated with a critical section and end speculation before performing the critical section; and providing output resulting from the synchronized parallel processes." (Emphasis added.) Thus, independent claim 16 should be allowable for at least reasons similar to those addressed above for claim 11.

Dependent claims 12-15 and 17-20 should be allowable based on their respective base claims and the additional recitations they contain. For example, claims 13 and 18 each recite, "wherein said effecting synchronization comprises translating at least one high-level software instruction into at least one machine instruction that controls speculative execution in a processor." The cited portion of Koseki (page 130, footnote 1) says nothing about a machine instruction that controls speculative execution in a processor. Thus, claims 13 and 18 should be allowable for at least this additional reason.

Independent claim 21 recites, "speculatively executing machine instructions, including a memory access instruction, in a processing system to effect synchronization between parallel processes, wherein the speculatively executing comprises performing a speculative read-modify-

write to a lock variable associated with a critical section; retiring the speculatively executed machine instructions to end speculation before performing the critical section; and maintaining cache coherence in the processing system during said executing and said retiring to identify a mis-speculation to effect the synchronization between the parallel processes." (Emphasis added.) The rejection of claim 21 should be withdrawn for at least reasons similar to those presented above with respect to claim 11. Moreover, nothing in Koseki describes maintaining cache coherence in a processing system to identify a mis-speculation to effect synchronization between the parallel processes, as claimed. Thus, claim 21 should be in condition for allowance.

Dependent claims 22-24 should be allowable based on their base claim and the additional recitations they contain. For example, claim 23 recites, "wherein said speculatively executing machine instructions comprises speculatively executing machine instructions in the processing system comprising multiple processors, and the mis-speculation comprises a memory dependency violation." (Emphasis added.) The "violation" in Koseki is one of one of misplaced lock reservation and has nothing to do with a memory dependency violation, which relates to reading and writing the same variable, as would be readily understood by those of ordinary skill in the art. Thus, claim 23 should be allowable for at least this additional reason.

Independent claim 25 recites, "a processor having a processor architecture that provides speculative execution of machine instructions and exposes said speculative execution to program control through at least one machine instruction; and a memory coupled with the processor, the memory embodying information indicative of instructions, including the at least one machine instruction, that result in synchronization between parallel processes when performed by the

processor with detection of mis-speculation; wherein performance of the instructions by the processor comprises performing a speculative read-modify-write to a lock variable associated with a critical section and ending speculation before performing the critical section." (Emphasis added.) Thus, independent claim 25 should be allowable for at least reasons similar to those addressed above for claims 11, 13 and 18. Dependent claims 26-30 and 32 should be allowable based on their base claim and the additional recitations they contain.

Independent claim 33 recites, "processing means for speculatively executing machine instructions in response to a speculative execution instruction, including means for detecting a mis-speculation; means for treating multiple speculative instructions as a group for purposes of retirement such that the multiple speculative instructions are flushed from the processing means together and execution proceeds from an address in response to a detected mis-speculation to effect synchronization between parallel processes; wherein performance of the instructions by the processing means comprises performing a speculative read-modify-write to a lock variable associated with a critical section and ending speculation before performing the critical section." (Emphasis added.) Moss fails to cure the noted deficiencies of Koseki. Moss teaches "lock-free" synchronization, which is mutually exclusive of using processor speculation in a data processing machine to speculatively read-modify-write a lock variable associated with a critical section, as claimed. For synchronization to be lock-free, there must be no lock. Thus, the proposed combination of references does not teach or suggest the claimed subject matter.

Moreover, to the extent that Koseki's lock reservation technique can be considered to teach a form of "speculation", which is not conceded, the "speculation" and detection of

"mis-speculation" in Koseki is all done in software. Nothing in Koseki suggests the use of hardware speculation, as the present means-for language ("processing means for speculatively executing machine instructions in response to a speculative execution instruction, including means for detecting a mis-speculation") necessarily entails. Thus, independent claim 33 should be allowable for at least reasons similar to those addressed above for claim 11 and for the additional language found in claim 33.

Dependent claim 34 should be allowable based on its base claim and the additional recitation it contains. Claim 34 recites, "wherein said means for detecting a mis-speculation comprises means for maintaining cache coherence in the processing means." The Office appears to refer to Koseki's "monitor cache" in rejecting this claim. However, the monitor cache in Koseki is a software data structure that is clearly distinct from the hardware cache referred to by claim 34. Thus, claim 34 should be allowable for at least this additional reason.

Conclusion

It is believed that all of the pending claims have been addressed. However, the absence of a reply to a specific issue or comment does not signify agreement with or concession of that issue or comment. Because the arguments made above may not be exhaustive, there may be reasons for patentability of any or all pending claims (or other claims) that have not been expressed. Finally, nothing in this paper should be construed as an intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any

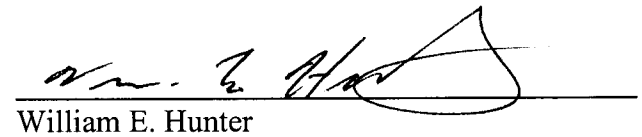
claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

It is respectfully suggested for all of these reasons, that the current rejections are overcome, that none of the cited art teaches or suggests the features which are claimed, and therefore that all of these claims should be in condition for allowance. A formal notice of allowance is thus respectfully requested.

No fees are believed to be due with this response. Nonetheless, please apply any necessary charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: Nov. 20, 2007



William E. Hunter
Reg. No. 47,671

Fish & Richardson P.C.
PTO Customer No. 20985
Telephone: (858) 678-5070
Facsimile: (858) 678-5099